REMARKS

Claims 1-4, 6-29 and 33 are currently pending in the subject application and are presently under consideration. Claims 1, 22, 25, 28 and 29 have been amended. It is respectfully submitted that no new subject matter has been added and the above amendments would not require any further search by Examiner.

The Examiner is thanked for courtesies extended during a telephonic interview held on July 7, 2008. Discussed were proposed amendments to claim 1 in light of the cited art. Although several features of the claims likely precluded the cited art, the Examiner felt a new search would be needed to verify patentability. As discussed, amending claim 1 to recite, in part, the map information store, the command information store and executables are stored separately, could traverse the currently cited art, especially in conjunction with a navigation component that modifies UI automation while maintaining compiled executables and that facilitates global variable replacement from a single location and sharing a common program flow. Accordingly, claim 1 above includes similar such amendments. Further, aspects of dependent claim 6 were discussed in view of the cited references. It was felt that employing information stored in a global information store when a global variable is encountered in the command information store likely traversed the art known to the Examiner. The Interview was conducted by Matthew Clapper (Reg. No. 62,216) and Examiner Augustine.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 1, 4-11, 14-22, 25, 27-29 Under 35 U.S.C. §103(a)

Claims 1, 4-11, 14-22, 25, 27-29 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Schaefer (US 2003/0084429) in view of Dewhurst *et al.* (US 6,430,609, hereinafter Dewhurst). Withdrawal of this rejection is requested since Schaefer and Dewhurst fail to teach or suggest all aspects of subject claims.

> A factfinder should be aware, of course, of the distortion caused by hindsight bias and must be cautious of arguments reliant upon *ex post* reasoning. See *KSR v. Teleflex*, 550 U.S. ___, 127 S. Ct. 1727 (2007) citing Graham v. John Deere Co. of Kansas City, 383 U. S. 1, 36

> (warning against a "temptation to read into the prior art the teachings of the invention in issue" and instructing courts to "'guard against slipping into the use of hindsight'" (*quoting Monroe Auto Equipment Co. v. Heckethorn Mfg. & Supply Co.*, 332 F. 2d 406, 412 (CA6 1964))).

Applicants' claimed innovation relates to system and methodology to facilitate navigation for inexperienced and experienced programmers to create user interface automation and to facilitate a modular system which can be modified without recompilation of the executables. To this end, independent claim 1 recites *a navigation component that facilitates simulated user interface associated with an automation component based, at least in part, upon information stored in a map information store and information stored in a command information store,* ***the map information store, the command information store and executables are stored separately, the navigation component modifies the user interface automation without recompiling executables by modifying one or more of the map information store or the command information store,*** *the navigation component further employs a global information store and facilitates a global variable replacement from a single location and sharing of a common program flow among a plurality of users.* Independent claim 22 recites *modifying the user interface automation by storing data, commands and executables separately and maintaining compilation of executables.* Independent claims 25, 28 and 29 recite similar aspects, specifically and in part, modifying a user interface automation by modifying a command file or map file and maintaining compilation of executables, and separately storing at least one of the map file, the command file or the compiled executables (claim 25). While the examiner stated that the aforementioned limitation was anticipated by the cited art (Schaefer and Dewhurst), applicants' representative would like to respectfully explain in further detail distinctions between the cited art and the subject claims (including previously presented and new distinctions).

Schaefer and Dewhurst neither teach nor suggest *the navigation component modifies the user interface automation by modifying at least one of the map information store or the command information store while maintaining compiled executables.* Significant processing overhead can be achieved by the navigation component by

employing previously compiled executables in modifying the user interface automation. By storing the map information store, command information store (*e.g.*, in a text file; see claims 3 and 4) and executables separately, map and/or command data can be updated without having to isolate executable information from the updates, or recompile the executable information. Accordingly, a synergy exists between modifying the user interface automation without recompiling executables and providing separate storage of map information, command information and executables. Simply providing distributed storage, for instance (*e.g.*, related to increased storage capacity or storage redundancy and security), would not obviate this synergy, because no special processing gains are available (*e.g.*, from maintaining previously compiled executables) with distributed storage; just the opposite is typically true, distributed storage often is associated with additional processing cost, not reduced processing. Further to the above, it should be appreciated that no aspect of the cited art discloses employing a global information store and facilitating global variable replacement from a single location and sharing of a common program flow among a plurality of users. Dewhurst is in sharp contrast to this, as user parameter updates are maintained specific to that user, enabling a user to run a remote application based on their inputs, independent of other user inputs. Schaefer is silent with respect to this recited feature, and thus cannot provide a common program flow among a plurality of users in conjunction with simulating a user interface and modifying the user interface based on an update of map or command information without recompiling executables. Discussion of the cited art and the further deficiencies of such art with respect to claim 1 and other claims is provided below.

Schaefer provides systems and methods for table driven automation testing for performing functional testing of a software program. The system includes a GUI translator component to translate one or more GUI maps into a set of database tables, a data input component to facilitate entry and editing of test case data in the tables, and a test engine component for executing the software program based on a test case stored in the tables. The Final Office Action dated June 2, 2008 acknowledges that the primary reference, Schaefer, does not teach the claimed innovation in its entirety with respect to *modifying the user interface automation without recompilation of executables by storing data, commands and executables separately, the navigation component further employs a*

global information store and facilitates a global variable replacement from a single location and sharing of a common program flow among a plurality of users and provides a secondary reference, Dewhurst, to compensate for the deficiencies of Schaefer. Dewhurst, given by Examiner, relates to a method for accessing complex software applications through a client user interface. The method includes accessing a master configuration file containing an array of configuration variables controlling the execution of software application, pre-selecting a subset of configuration variables from the array of configuration variables contained in the master configuration file, generating the client user interface to modify only the subset of configuration variables and transmitting the client user interface to client computer. Nowhere does this reference teach *modifying the user interface automation without recompilation of executables by storing data, commands and executables separately*, however. Further, nowhere does Dewhurst teach *facilitating global variable replacement from a single location and sharing of a common program flow among a plurality of users*.

As a more specific summary to illustrate the deficiencies of Dewhurst, Dewhurst provides for a method for configuring and executing a software application with a client user interface. The method includes accessing a master configuration file containing an array of configuration variables controlling the execution of software application, pre-selecting a subset of configuration variables from the array of configuration variables contained in the master configuration file, generating the client user interface to modify only the subset of configuration variables, transmitting the client user interface to a client computer, receiving the subset of configuration variables as modified on the client computer, applying the subset of configuration variables against the master configuration file, executing the software application in accordance with the master configuration file on a computational server, the execution of software application producing an output result, and transmitting the output result to the client computer (*See*, Col. 4, lines 33-65). An expert user generates a template file that specifies a pre-configured engineering simulation of two pipes intersecting. The template file is rendered on the client computer as a graphic user interface. *The novice user may change the default values displayed in the graphic user interface and then directly transmit the new values to the computational server*. The results are then accessed by the client computer. In order to effectively

handle overlapping job requests for a plurality of client computers, a queue is established *to hold job requests until the computational server is available to process them* (*See*, Col. 11, lines 6-47). The newly modified master configuration file by the novice user is *then transmitted to the computational server which then executes the application and returns the results to the end user* (*See*, Col. 12, lines 26-30).

Hence, Dewhurst allows novice users to utilize a complex software application for engineering and financial tasks. An expert user anticipates simulations or reports commonly required by novice users and generates a library of template files suitable for those simulations/reports. The template files comprise a subset of configuration variables pre-selected by the expert user from an array of configuration variables contained in a master configuration file. The novice users access the template files generated by the expert user and change default values, if required, displayed in the graphic user interface *and transmit the new values to a computational server. The computational server then executes the application according to new values and returns the result* ***to the end user***. More particularly, Dewhurst requires transmitting the new values (or changed values for configuration variable) *to a computational server and executing the application again for returning the result according to new values set by the novice user. Hence executables, changed by the novice user, are executed again with the application at the computational server.* Dewhurst does not contemplate *modifying the user interface automation without recompilation of executables or while maintaining compiled executables, as recited in the subject independent claims.* Further, nowhere does Dewhurst teach or suggest *storing data, commands and executables separately.* ***Through this feature, the claimed subject matter facilitates a modular system which can be modified quickly and efficiently without recompilation of the executables***. A beneficial synergy is obtained by storing updateable information separate from executables that are maintained in a compiled form. For instance, the updateable information can be stored in simple text files separate from executables associated with program flow to mitigate a likelihood that that executables need to be modified and/or recompiled based on changes to the updateable information. Thus, modification of the text files ***produces new behavior*** for the claimed systems and methods; and ***update to the program flow only requires a modification to these text files and not the executables or engine***.

At page 4 of Final Office Action, it is erroneously asserted that Dewhurst teaches *the navigation component employs a global information store and facilitates a global variable replacement from a single location and sharing of a common program flow among a plurality of users*, with respect to independent claim 1. Dewhurst provides for an expert user generating a library of template files wherein *a subset of configuration variables are pre-selected by the expert user* from an array of configuration variables contained in a master configuration file. Novice users access the required template file generated by the expert user and *modify only the subset of configuration variables* and transmit the new values to a computational server for execution (*See*, Col. 11, lines 6-47). An end-user, as provided by Dewhurst, *cannot modify all variables in the array of configuration variables contained in the master configuration file* and can only modify the subset of configuration variables which were pre-selected by the expert user into the template file. Further, the subset of variables changed by one user is executed at a computational server and results are transmitted to *that user*, and ***do not result in a common program flow shared among all users***.

With respect to dependent claims 6, Final Office Action erroneously asserts (at p. 6) that Schaefer teaches *the navigation component employing information stored in the global information store when a global variable is encountered in the command information store*. Schaefer provides for testing certain aspects of a software program. Therefore, a user only generates GUI maps for aspects of the software program which the user desires to test. For example, if the user desires to only test the Flight Reservations window, the user generates a single GUI map having a filename "Flight Reservations.GUI". The window object has the logical name "Flight Reservation" and its class type is "window". In addition, the GUI map for the Flight Reservation window includes object information for the other objects on the window, including "Date of Flight" text field, "Flights" button, "Fly From" pull down list, and "Fly To" pull down list (*See*, Paragraphs [0058] & [0060]). However Schaefer *et al.* does not contemplate a navigation component employing global information upon encountering a global variable in a command information store. Thus, a user can update the command information store with a global variable and the navigation component can, for instance, automatically access a global information store and employ information contained therein. The cited

references are completely silent with respect to this aspect of claim 6. Hence, it is submitted that one of ordinary skill in the art would not be motivated to modify the disclosures of the cited art to arrive at this claimed aspect, particularly considering the silence of the cited art on this matter.

In view of at least the foregoing, it is clear that Schaefer and Dewhurst fail to teach or suggest to one of ordinary skill in the art each and every aspect recited in independent claims 1, 22, 25, 28 and 29, and dependent claims 4-11, 14-21, and 29. Therefore, it is respectfully requested that this rejection be withdrawn.

II. Rejection of Claim 2 Under 35 U.S.C. §103(a)

Claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Schaefer in view of Dewhurst in further view of Minard (US 6,247,020). It is respectfully requested that this rejection be withdrawn for at least the following reasons. Schaefer, Dewhurst and Minard either alone or in combination do not teach or suggest all aspects set forth in the subject claims. Minard relates to development system with application browser user interface and does not make up for the aforementioned deficiencies of Scharefer and Dewhurst with respect to *the navigation component modifying the user interface automation without recompiling executables by modifying the map information store and/or the command information store,* as recited in independent claim 1 (from which claim 2 depend). Thus it is submitted, the subject innovation as recited in claim 2 is not obvious over the combination of Schaefer, Dewhurst and Minard. Accordingly, it is respectfully submitted that this rejection should be withdrawn.

III. Rejection of Claims 12-13, 23 and 26 Under 35 U.S.C. §103(a)

Claims 12-13, 23 and 26 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Schaefer in view of Dewhurst in further view of Zimniewiez *et al.* (US 6,744,450). It is respectfully requested that this rejection be withdrawn for at least the following reasons. Scharefer, Dewhurst and Zimniewiez *et al.* either alone or in combination do not teach or suggest all aspects set forth in the subject claims. Zimniewiez *et al.* relates to system and method for providing multiple installation actions

and does not make up for the deficiencies of Scharefer and Dewhurst with respect to *the navigation component modifying the user interface automation without recompiling executables by modifying the map information store and/or the command information store,* as recited in independent claims 1 (from which claim 12 and 13 depend), 22 (from which claim 23 depends) and independent claim 25 (from which claim 26 depends). Thus it is submitted, the subject innovation as recited in claims 12, 13, 23 and 26 is not obvious over the combination of Scharefer, Dewhurst and Zimniewicz *et al.* Accordingly, it is respectfully submitted that this rejection should be withdrawn.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Himanshu S. Amin/
Himanshu S. Amin
Reg. No. 40,894

AMIN, TUROCY & CALVIN, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731